Unmasking Fault Tolerance: Masking vs. Non-masking Fault-tolerant Systems

Nils Müllner
nils.muellner@informatik.uni-oldenburg.de

System Software and Distributed Systems Group,
Universität Oldenburg, Germany

June 30, 2009

Table of contents

AVACS

- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

AVACS

- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

AVACS

- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

AVACS

- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

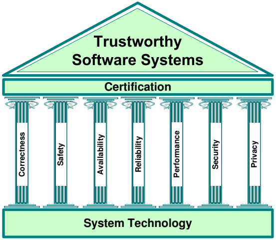
AVACS

- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

AVACS

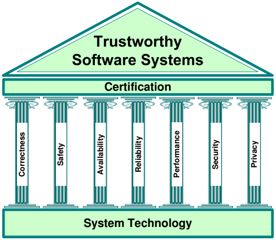
- Automatic Verification and Analysis of Complex Systems
- transregional: 3 Universities, $3 * 4$ years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)

TrustSoft



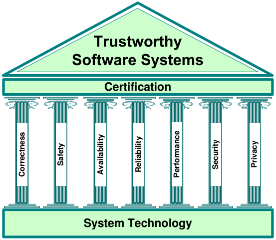
- Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)
- Fault Tolerance Metrics = Availability, Reliability
- founded April 1st 2005
- 13 supervisors, 8 students (8 finished, 14 scholarships)

TrustSoft



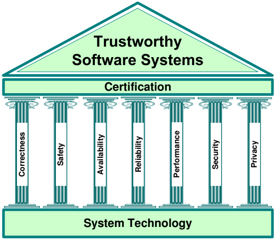
- Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)
- Fault Tolerance Metrics = Availability, Reliability
- founded April 1st 2005
- 13 supervisors, 8 students (8 finished, 14 scholarships)

TrustSoft



- Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)
- Fault Tolerance Metrics = Availability, Reliability
- founded April 1st 2005
- 13 supervisors, 8 students (8 finished, 14 scholarships)

TrustSoft



- Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)
- Fault Tolerance Metrics = Availability, Reliability
- founded April 1st 2005
- 13 supervisors, 8 students (8 finished, 14 scholarships)

Thesis Topics

- MSc: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures
    - distributed systems
    - self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
    - transient faults
- PhD: Unmasking Fault Tolerance

Thesis Topics

- MSc: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures
  - distributed systems
  - self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
  - transient faults
- PhD: Unmasking Fault Tolerance

Thesis Topics

- MSc: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures
  - distributed systems
  - self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
  - transient faults
- PhD: Unmasking Fault Tolerance

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	¬ **safe**
live	masking	nonmasking
¬ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	¬ **safe**
live	masking	nonmasking
¬ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	¬ **safe**
live	masking	nonmasking
¬ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	¬ **safe**
live	masking	nonmasking
¬ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	¬ **safe**
live	masking	nonmasking
¬ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

	safe	$\neg$ **safe**
live	masking	nonmasking
$\neg$ **live**	fail safe	-

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?

Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated yet

- how much time/space per "quality of service"?

Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?

Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?

Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

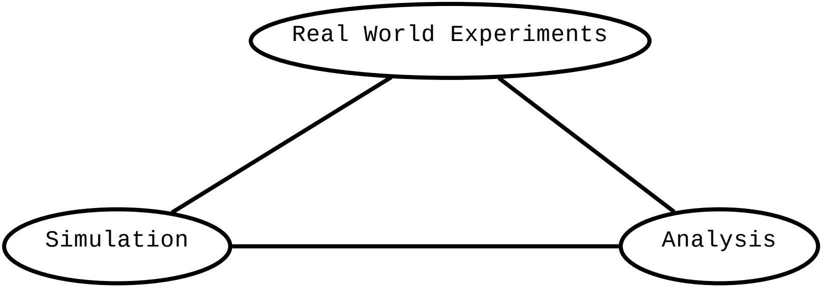
- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?

Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing
- tradeoff between degree of masking vs. cost not investigated, yet
- how much time/space per "quality of service"?

Scientific Triangle



Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Methods for Derivation

- formal analysis ⇒ state space explosion
- simulation ⇒ inaccurate, no proof
- real world experiments ⇒ expensive

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc ⇒ 6561 states, 5 proc ⇒∼ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc ⇒ 6561 states, 5 proc ⇒∼ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow \sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

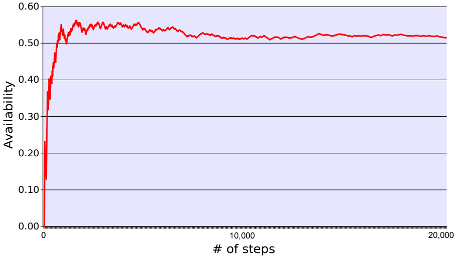
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Formal Analysis

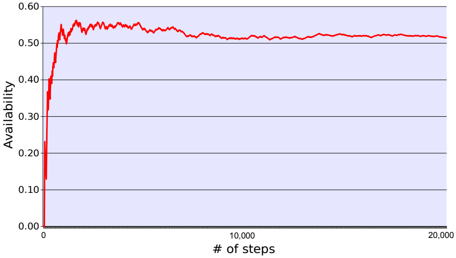
- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow\sim$ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability
- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Simulation



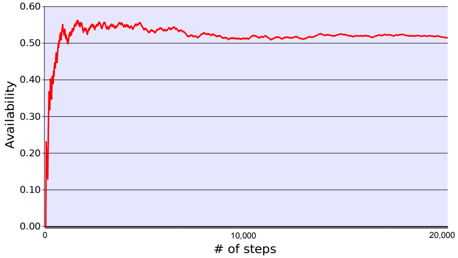
- based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to $n + 2$ cores (nodes + fault-injector + server)

Simulation



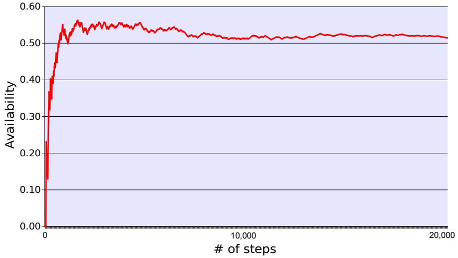
- based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to $n + 2$ cores (nodes + fault-injector + server)

Simulation



- based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to $n + 2$ cores (nodes + fault-injector + server)

Simulation



- based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to $n + 2$ cores (nodes + fault-injector + server)

Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive ($\sim$200 AUD/mote, $\sim$100 motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  - hence: derivation of good solutions by analysis & simulation before the testing in real world should save money, time

Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
    - too expensive (~200 AUD/mote, ~100 motes for a realistic network)
    - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
    - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time

Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive (∼200 AUD/mote, ∼100 motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time

Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive (∼200 AUD/mote, ∼100 motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time

Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
- result: manageable set of close-to-optimal solutions
- then test given results in real world for real measures

Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
- result: manageable set of close-to-optimal solutions
- then test given results in real world for real measures

Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
- result: manageable set of close-to-optimal solutions
- then test given results in real world for real measures

Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
- result: manageable set of close-to-optimal solutions
- then test given results in real world for real measures

Results So Far

- simulation and analysis deliver similar results (two papers)

- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)

- definition of new metrics (Instantaneous Window Availability)

Results So Far

- simulation and analysis deliver similar results (two papers)
- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)
- definition of new metrics (Instantaneous Window Availability)

Results So Far

- simulation and analysis deliver similar results (two papers)
- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)
- definition of new metrics (Instantaneous Window Availability)

Results So Far

- simulation and analysis deliver similar results (two papers)

- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)

- definition of new metrics (Instantaneous Window Availability)

Results So Far

- simulation and analysis deliver similar results (two papers)
- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)
- definition of new metrics (Instantaneous Window Availability)

Instantaneous Window Availability

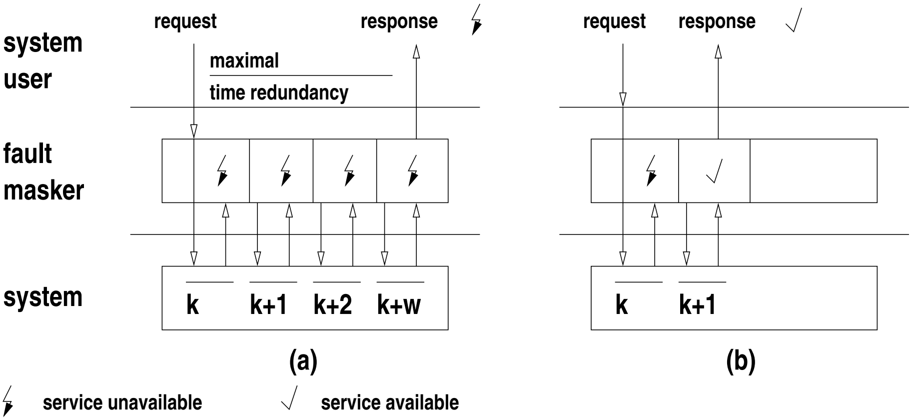
- for measuring relation/tradeoff between cost (time) and degree of masking
- to be extended: IW reliability, maintainability,...
- to be extended: relation of cost (space) and degree of masking

Instantaneous Window Availability

- for measuring relation/tradeoff between cost (time) and degree of masking
- to be extended: IW reliability, maintainability,...
- to be extended: relation of cost (space) and degree of masking

Instantaneous Window Availability

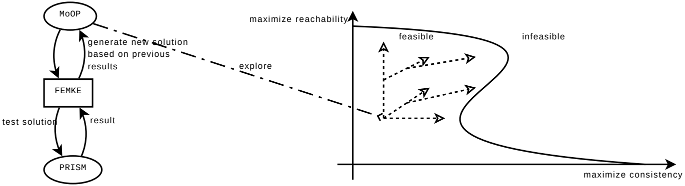
- for measuring relation/tradeoff between cost (time) and degree of masking
- to be extended: IW reliability, maintainability,...
- to be extended: relation of cost (space) and degree of masking



system
user
fault
masker
system
request
response
maximal
time redundancy
k
k+1
k+2
k+w
(a)
request
response
k
k+1
(b)
service unavailable
service available

Work In Progress

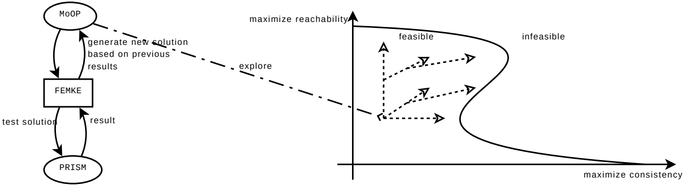
- automatic computation of Pareto-optimal solution sets



- derivation of trade-off between availability, consistency and redundancy (space) for WSNs

Work In Progress

- automatic computation of Pareto-optimal solution sets



- derivation of trade-off between availability, consistency and redundancy (space) for WSNs

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, *satisfactory* solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, *satisfactory* solutions can be derived in a reasonal amount of time, if exact calculation takes too long

Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, *satisfactory* solutions can be derived in a reasonal amount of time, if exact calculation takes too long

- A. Dhama, O. E. Theel, and T. Warns, "Reliability and Availability Analysis of Self-Stabilizing Systems," in *SSS '06*, 2006, pp. 244–261.
- A. Avizienis, J.-C. Laprie, B. Randell, and V. M. U, "Fundamental Concepts of Dependability," in *Proceedings of the Third Information Survivability Workshop*, 2000.
- S. Dolev, *Self-Stabilization*. Cambridge, MA, USA: MIT Press, 2000.
- M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic Model Checking for Performance and Reliability Analysis," *ACM SIGMETRICS Performance Evaluation Review*, vol. To Appear, 2009.
- N. Müllner, A. Dhama, and O. Theel, "Derivation of Fault Tolerance Measures of Self-Stabilizing Algorithms by Simulation," in *ANSS '08*, Ottawa, Ontario, Canada, 2008, pp. 183–192.
- E. W. Dijkstra, "Guarded Commands, Nondeterminacy and Formal Derivation of Programs," *Comm. ACM*, vol. 18, no. 8, pp. 453–457, 1975. [Online]. Available: `http://dx.doi.org/10.1145/360933.360975`

Bonus: fun with weakening the masking property 1/3

- self-stabilizing systems are nonmasking fault tolerant
- during stabilization/repair phase, user is exposed to faults
- why not even weaken nonmasking self-stabilizing systems?
- unmask fault tolerance even below nonmasking...

Bonus: fun with weakening the masking property 1/3

- self-stabilizing systems are nonmasking fault tolerant
- during stabilization/repair phase, user is exposed to faults
- why not even weaken nonmasking self-stabilizing systems?
- unmask fault tolerance even below nonmasking...

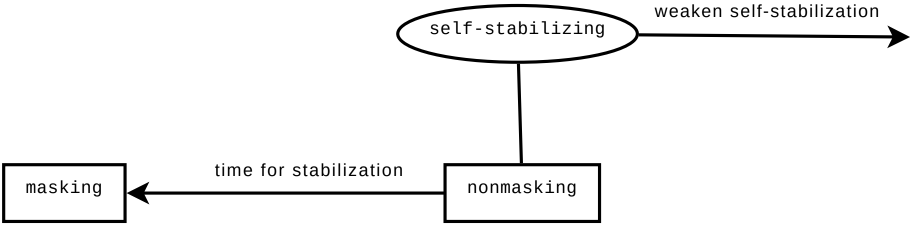
Bonus: fun with weakening the masking property 1/3

- self-stabilizing systems are nonmasking fault tolerant
- during stabilization/repair phase, user is exposed to faults
- why not even weaken nonmasking self-stabilizing systems?
- unmask fault tolerance even below nonmasking...

Bonus: fun with weakening the masking property 1/3

- self-stabilizing systems are nonmasking fault tolerant
- during stabilization/repair phase, user is exposed to faults
- why not even weaken nonmasking self-stabilizing systems?
- unmask fault tolerance even below nonmasking...

Bonus: fun with weakening the masking property 2/3



Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Bonus: fun with weakening the masking property 3/3

- convergence: $\exists t_k : x(t_k) \models P$
- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$
- attack at convergence: slow or probabilistic
- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$

Attack Convergence

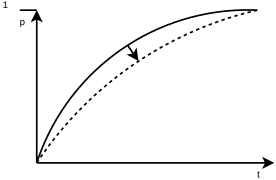


Figure: slower stabilization

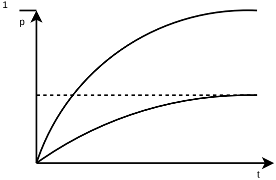
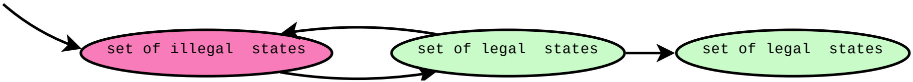


Figure: probabilistic stabilization

Attack Closure



Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?

Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?

Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?